

# Genetic algorithm based technique for solving chance constrained problems<sup>★</sup>

Chandra A. Poojari<sup>\*</sup> and Bobby Varghese

*Centre for the analysis of Risk and optimisation modelling applications  
(CARISMA), School of Information systems, computing and Mathematics, Brunel  
University*

---

## Abstract

Management and measurement of risk is an important issue in almost all areas that require decisions to be made under uncertain information. Chance Constrained Programming (CCP) have been used for modelling and analysis of risks in a number of application domains. However, the resulting mathematical problems are non-trivial to represent using algebraic modelling languages and pose significant computational challenges due to their non-linear, non-convex, and the stochastic nature. We develop and implement C++ classes to represent such CCP problems. We propose a framework consisting of Genetic Algorithm and Monte-Carlo simulation in order to process the problems. The non-linear and non-convex nature of the CCP problems are processed using Genetic Algorithm, whereas the stochastic nature is addressed through simulation. The computational investigations have shown that the framework can efficiently represent and process a wide variety of the CCP problems.

*Key words:* Chance constrained models, Genetic algorithm, Simulation, Stochastic Programming

---

## 1 Background to Stochastic Programming

Addressing data uncertainty in mathematical programming models has been a central problem in optimization since a long time. There are two principal methods that have been proposed to address data uncertainty over the

---

<sup>★</sup> The work was supported by Brunel University's BRIEF award. The insightful comments of Prof. John Beasley to the earlier drafts has helped in improving the content and the format of this paper.

<sup>\*</sup> Chandra.Poojari@brunel.ac.uk

years: (a) Stochastic Programming(SP) (Dantzig[13], Beale[5], Charnes and Cooper[11]), and (b) Robust Optimization (RO) (Ben-Tal and Nemirovski [6], El-Ghoui and Lebret[22]).

SP models develop plans that are evaluated against a variety of future possibilities that represent the alternative outcomes of the data. Such models yield plans that are better able to hedge against losses and catastrophic failures. Because of these properties SP models have been developed for a variety of applications, including electric power generation [42], financial planning [15,19,68], telecommunication network planning [57,65], supply chain management [40], Oil industry [16], vehicle manufacturers [21], electricity suppliers [49,64], environment [35], transportation [36,20], construction, energy, chemical processing [31], aerospace, and military system [63].

Stochastic programming models can include both *anticipative* and *adaptive* decision variables. Anticipative variables corresponds to those decisions that must be made *here-and-now* and cannot depend on the future observations/partial realizations of the random parameters. Adaptive variables corresponds to *wait-and-see* decisions after some/all of the random parameters are observed.

Consider

- $\mathbf{x} = \{x_1, x_2, \dots, x_n\}$  as the decision vector,  $x \in \mathbb{R}^n$ ,
- $\xi = \{\xi_1, \xi_2, \dots, \xi_m\}$  as the stochastic vector,  $\xi \in \mathbb{R}^m$ , having the expected value vector  $\mu_\xi$  and the variance-covariance matrix  $\sigma_\xi^2$ .
- $f(\mathbf{x}, \xi)$ ,  $g_i(\mathbf{x})$ ,  $g_i(\mathbf{x}, \xi)$ ,  $h_i(\mathbf{x})$  and  $h_i(\mathbf{x}, \xi)$  as real valued functions.  $f(\mathbf{x}, \xi): \mathbb{R}^{n+m} \rightarrow \mathbb{R}$  is the objective function, and  $g_i(\mathbf{x})$  &  $h_i(\mathbf{x}): \mathbb{R}^n \rightarrow \mathbb{R}$  are the deterministic  $g_i(\mathbf{x}, \xi)$  &  $h_i(\mathbf{x}, \xi): \mathbb{R}^{n+m} \rightarrow \mathbb{R}$  are the stochastic constraints.
- $E_\xi[\cdot]$  denote the expectation operator with respect to  $\xi$ ,
- $Pr[\cdot]$  denote the probability operator.

Consider the Stochastic program

$$\begin{aligned}
 P_{SP} \quad & \text{Max } f(\mathbf{x}, \xi) \\
 & \text{Subject to} \\
 & g_i(\mathbf{x}) \leq 0 \quad i = 1, 2, \dots, p_1 \quad (1) \\
 & g_i(\mathbf{x}, \xi) \leq 0 \quad i = p_1 + 1, p_1 + 2, \dots, p \quad (2) \\
 & h_i(\mathbf{x}) = 0 \quad i = 1, \dots, q_1 \quad (3) \\
 & h_i(\mathbf{x}, \xi) = 0 \quad i = q_1 + 1, q_1 + 2, \dots, q \quad (4)
 \end{aligned}$$

Let  $G^d$ ,  $G^s$ ,  $H^d$ , and  $H^s$  denote the index sets for the constraints 1, 2, 3, and 4 respectively. Therefore  $|G^d| = p_1$ ,  $|H^d| = q_1$ ,  $|G^s| = p - p_1$ ,  $|H^s| = q - q_1$ .

Once the values of the functions of a mathematical program are random as in  $P_{SP}$ , it is not immediately clear how to formulate a well-posed problem. A

number of possible formulations of the SP model ( $P_{SP}$ ) are possible depending on when decisions must be taken relative to the realisation of the random variables, the degree to which the constraints structure must be satisfied, and the choice of a stochastic optimality criterion, that is, an objective or return function.

In an adaptive model, information related to uncertainty  $\xi$  becomes partially available before decision making, so optimization takes place in a learning environment. An adaptive stochastic program can be formulated as:

$$\begin{aligned}
P_{SP-Adap} \quad & \text{Max} \quad (f(\mathbf{x}, \xi)|\xi) \\
& \text{Subject to} \\
& g_i(\mathbf{x}) \leq 0 \quad i \in G^d \\
& (g_i(\mathbf{x}, \xi)|\xi) \leq 0 \quad i \in G^s \\
& h_i(\mathbf{x}) = 0 \quad i \in H^d \\
& (h_i(\mathbf{x}, \xi)|\xi) = 0 \quad i \in H^s.
\end{aligned}$$

A generic anticipative stochastic programming problem can be formulated as follows:

$$\begin{aligned}
P_{SP-Anti} \quad & \text{Max} \quad E_\xi[f(\mathbf{x}, \xi)] \\
& \text{Subject to} \\
& g_i(\mathbf{x}) \leq 0 \quad i \in H^d \\
& E_\xi[g_i(\mathbf{x}, \xi)] \leq 0 \quad i \in G^s \\
& h_i(\mathbf{x}) = 0 \quad i \in H^d \\
& E_\xi[h_i(\mathbf{x}, \xi)] = 0 \quad i \in H^s.
\end{aligned}$$

Anticipative models can either be formulated at Stochastic Programming with Recourse (SPR) or as Chance Constrained Programming (CCP) models.

In the SPR formulations, the decision variable can be considered as a tuple  $(x_1, x_2(\xi))$ , where  $x_1 \in \mathbb{R}^{n_1}$  and  $x_2 \in \mathbb{R}^{n_2}$ , ( $n = n_1 + n_2$ ). Some of these decisions, that is  $x_1$ 's, are made prior to the resolution of the uncertainty and the remaining decisions,  $x_2$ 's, are made based on the realised value of the uncertainty parameters. The objective function consists of two components. The first component is the cost for making the *here-and-now* decisions,  $x_1$ , and the second component is the expected value of the penalty for taking corrective actions,  $x_2(\xi)$ . Algorithms for processing SPR usually exploit the mathematical and the problem structures. Primal and dual decomposition based techniques have been extensively researched [62,8,51,41]. In addition approximation techniques are used to represent the sample space of the random variables [52,60,61,50,27].

In the CCP formulation, we would require that a particular variable/function lie within a target range with a certain probability. CCP models often lead to nonlinear, even non-convex, constraints. Early computational implementation of CCP problems was restricted to normally distributed random variables. Prékopa [44] showed that a much larger class of random variables yield the convexity property. He proved that if the constraint functions are linear/affine in  $\mathbf{x}$  and randomness only appears additively, and the random variable has a log-concave probability density function, then the resulting feasible region is convex. For discrete random variables this convexity property no longer holds, however, the set of feasible solutions can be represented as a disjunctive set [56]. Prékopa [45] proposed a dual type algorithm for solving the general chance constrained problem. In [46], he discusses several approaches for solving chance-constrained models including gradient methods and the use of penalty functions. Hiller [28] proposed a procedure for approximating chance constraints by linear constraints. He replaced the chance constraints by the separable convex constraints and then solved the problem by the separable convex programming technique using the simplex algorithm. Seppala [58] constructed sets of uniformly tighter linear constraints that replaces a single chance constraint. According to Seppala this method is more accurate, but less efficient than Hiller's procedure. The method is not restricted to bounded variable type problems, however, it requires many more constraints than Hiller's method. Seppala and Orpana [59] examined the efficiency and accuracy of the algorithm developed by Seppala. Olson and Swenseth [43] proposed an approximation formula for solving CCP problems that places bound on the chance constraints which is at least as tight as the true non-linear form. This technique, however, is highly inaccurate (see Yahia and Daneshmand [71]). Ishii et al. [32] proposed a new algorithm for solving a class of the CCP problem that realises the difficulty in determining the risk level beforehand. For this purpose, the computation of risk level for each chance constraint was made a part of the solution process. Weintraub and Vera [70] proposed a cutting plane approach for solving the deterministic equivalent of a CCP for the case of normally distributed random constrained coefficients. Dentcheva, Prékopa and Ruszcynski [17] have define a partial order on the set of scenarios and show that a limited number of scenarios (called  $(1 - \alpha)$  efficient points) play a key role. They develop an algorithm that iteratively updates the set of efficient points to generate a tight lower and upper bound. The algorithm has been extended to general convex programming with probabilistic constraints in [18]. Beraldi and Ruszcynski [7] have discussed methods based on partial or complete enumeration of efficient points for stochastic integer programming. Ruszcynski [53] reformulates the CCP problems as large-scale Mixed Integer Programming problems with knapsack constraints and develop valid inequalities for the MIP problem. Aringheri[2] develops a Tabu search and simulation for two processing telecommunication network optimisation problems formulated as CCP. Our work is inspired by the work done by Iwamura and Liu [33] who proposed a stochastic simulation based genetic algorithm in which the

stochastic simulation is used to check the chance constraints and to calculate the goal functions.

## 2 Contribution of the paper

There is an increasing interest in the use of mathematical programming models to quantify and measure risks in diverse industrial sectors. The non-linear nature of the utility function, and the non-convex optimisation problem having discrete decision variables requires optimisation scientist and researchers to combine alternate tools and techniques in order to develop efficient computational techniques. In this paper we have proposed and developed a generic framework combining Genetic algorithm and simulation.

The proposal and adoption of the stochastic mathematical programming input data standard, [9], as well as the library of models maintained by researchers [3,10] prepared to this specification, has made it easier to develop solvers for processing SPR models. One reason for lack of significant computational work in processing CCP problems is the absence of suitable representation scheme for benchmark models, therefore it is difficult to evaluate the performances of new techniques on a wide variety of CCP problems. We are not aware of any earlier research proposing and investigating a generic framework in order to represent and process CCP problems. To the best of our knowledge this is the first instance where C++ classes have been developed and implemented to represent the alternative CCP formulations. Also, we are able to represent deterministic and stochastic optimisation problem, that can have multiple objective functions and linear/ non-linear variables and constraints. Moreover, the variables could continuous or discrete.

A parameter-less penalty function is developed in order to process constrained optimisation problems using GAs. In addition, we define fitness functions that scores a given solution based on its feasibility and optimality. The fitness function is designed to provide greater emphasis on optimality during the initial generations of the GA, so as to avoid the algorithm to converge to the local optima. However, the priority given to the optimal solutions is gradually decreased during the evolution and more preference is given to the feasible solutions at higher generations. Statistical analysis are carried out to investigate the performances of the penalty and the fitness functions.

The structure of the paper is as follows. In section 3 we discuss the alternate formulations of a CCP model. In section 4 we provide a brief overview of Genetic Algorithm. In section 5 we discuss the computational framework combining GAs and Monte-carlo based simulation. In section 6 we discuss the performance of the framework on set of test problems. Section 7 covers dis-

cussion and conclusion.

### 3 Alternate formulations of CCP models

Chance constrained programming (CCP)[11] is an anticipative stochastic optimization approach, which can deal explicitly with randomness. Charnes and Cooper suggested 3 different types of objective functions,

- (1) a function that maximizes the expected value of the objective function (the **E**-model),

$$\text{Max} \quad E_{\xi} [f(\mathbf{x}, \xi)]$$

- (2) a function that minimizes the generalized mean square of the objective function (the **V**-model),

$$\text{Min} \quad [f(\mathbf{x}, \xi) - E_{\xi} [f(\mathbf{x}, \xi)]]^2$$

and,

- (3) a function that maximizes the probability of satisfying an aspiration level of the objective function (the **P**-model),

$$\text{Max} \quad Pr \left[ f(\mathbf{x}, \xi) \{ \underset{\sim}{\geq} \} u \right]$$

where  $u$  is a specified risk level.

In CCP, the focus is on the system's ability to meet the constraints (risk measures) with certain reliability in an uncertain environment. These stochastic constraints can be represented in three alternative ways.

- (1) Satisfying the expected value of constraints.

$$\left. \begin{array}{ll} E_{\xi}[g_i(\mathbf{x}, \xi)] \leq 0, & i \in G^s \\ E_{\xi}[h_i(\mathbf{x}, \xi)] = 0, & i \in H^s \end{array} \right\} \text{expectation constraint form} \quad (5)$$

Let  $\bar{G}^s$  and  $\bar{H}^s$  denote the set of constraints whose expected value need to be satisfied,

$$\bar{G}^s = \{g_i | E_{\xi}[g_i(\mathbf{x}, \xi)] \leq 0\} \quad (6)$$

$$\bar{H}^s = \{h_i | E_{\xi}[h_i(\mathbf{x}, \xi)] = 0\}. \quad (7)$$

(2) Satisfying the constraints probabilistically.

Define a set consisting of the combination of  $r$  ( $\leq p_1$ ) stochastic inequality constraints as  $\hat{g}_{i,r}(\mathbf{x}, \xi)$  ( $|\hat{g}_{i,r}| = r$ ), defined as

$$= \{g_{i_1}, g_{i_2}, \dots, g_{i_r} | i_1, i_2, \dots, i_r \in G^s \text{ and } i_1 \neq i_2 \dots \neq i_r\}. \quad (8)$$

The set consisting of all unique possible combinations of  $r$  stochastic inequality constraints is denoted by  $\hat{G}_r^s$ , defined as

$$= \{\{\hat{g}_{i,r}\}, \{\hat{g}_{j,r}\}, \dots\}. \quad (9)$$

The number of sets within the set  $\hat{G}_r^s$  is equal to  $\mathcal{C}_r^{p_1} (= \frac{p_1!}{(p_1-r)!r!})$ .

The probabilistic constraints are given by

$$Pr[\hat{g}_{i,r}(\mathbf{x}, \xi) \leq 0] \geq \alpha_{i,r}, \quad \hat{g}_{i,r}(\mathbf{x}, \xi) \in \hat{G}_r^s \quad (10)$$

where  $\alpha_{i,r}$  is the minimum probability measure that the constraint sets are required to satisfy. Based on the value of  $r$ , we can interpret the constraint 10 in two different ways.

(a) When the value  $r = 1$ , the constraint 10 reduces to individual chance constraints of the form

$$Pr[\hat{g}_{i,1}(\mathbf{x}, \xi) \leq 0] \geq \alpha_{i,1}, \quad \hat{g}_{i,1}(\mathbf{x}, \xi) \in \hat{G}_1^s \quad (11)$$

which can be further interpreted as

$$Pr[g_i(\mathbf{x}, \xi) \leq 0] \geq \alpha_i, \quad i \in G^s \quad (12)$$

where  $\alpha_i$  is the the minimum probability measure (reliability/safety index) for each constraint  $i$ .

(b) On the other hand, when  $r > 1$ , the constraint 10 reduces to joint chance constraints. Such constraints arise when we require that a group of stochastic constraints satisfy a given reliability level.

It is well-known that  $Pr\{\xi = y\} = 0, \forall y \in \mathcal{R}^m$ , therefore equality probabilistic constraints do not have any mathematical justification. Let  $\Gamma$  denote the set of constraints in the CCP problem,  $\Gamma$  includes the deterministic, the expected value (6 and 7) and the probabilistic constraints 10.

A general CCP problem can involve an objective function with several linear/nonlinear, joint/individual probabilistic constraints. The resulting optimisation problem can be classified as a nonlinear programming problem where the nonlinear constraint involve multidimensional integrals. The computation of the probabilistic constraint and its gradient involves Monte Carlo integration, which results in inaccurate function values and gradients. Moreover, the objective function and the constraints may also be nonconvex. An additional source of severe numerical difficulties has its roots in the nature of probability

distribution functions: the magnitude of the components of the gradient is usually very small outside a narrow region.

## 4 An overview of Genetic Algorithm

GAs have been applied to optimization problems in many fields, such as optimal control problems, job scheduling, transportation problems, pattern recognition, machine learning [38,23,37,29]. GAs are robust algorithms that are capable of optimizing multi-modal, noisy, dynamic functions.

GAs start out with an initial *population* of possible solutions, called individuals, to a given problem where each individual is represented using some form of encoding as a *chromosome*. These chromosomes are evaluated for their *fitness* -that is the extent to which the individuals they represent are suitable to the environment. Based on their fitness as a criterion, certain chromosomes in the population are selected for reproduction. These selected individuals are manipulated by *crossover* and *mutation* operators. The crossover operator is applied to a pair of selected parents to create offspring, and the mutation operator is used as a slight modification to this offspring, or of the remaining members of the population. The repeated application of the genetic operators to the fittest chromosomes results in an increase in the average fitness of the population over time and thus to the identification of improved solutions to the problem under investigation. Each new generation of solutions is seen to be in some sense an improvement over the previous one. This process is iteratively repeated until there appears in the population an acceptable solution to the problem at hand.

GAs have been extremely successful in solving unconstrained optimisation problems. Several methods had been proposed to handle constraints within GAs for numerical optimisation problems [30,34,54,39,14,4,12,48,37,47].

## 5 Computational Framework Combining GA and Monte Carlo Simulation

In this paper, CCP optimisation problems are processed by extending the Genetic Algorithm library of Matthew Wall [69] for constrained optimisation problems. The random variables  $\xi$ 's are generated from their underlying distribution using the random number generator of Robert Davies [1]. Figure 1 shows the graphical representation of the framework.



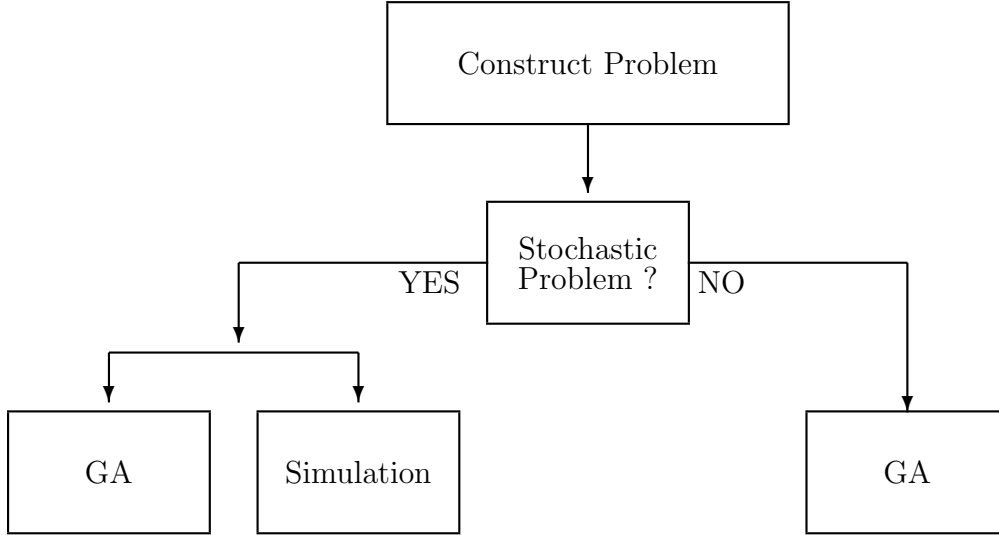


Fig. 1. Optimization framework using GA and Simulation

### 5.1 Representation of the Problem

The underlying CCP problem is created using C++ classes. These classes are versatile to represent the alternate CCP problems discussed in section 3. The classes represent the various components of an optimisation problem such as the objective function(s), the constraint(s), and the variable(s).

#### *Objective function*

The class for the objective function can represent single and multiple functions. These functions can either be deterministic and/or stochastic. Furthermore, the stochastic objective functions could involve the evaluation of either probabilistic and/or the expected value.

#### *Constraints*

The constraint class is able to represent deterministic and stochastic constraints. Within the stochastic constraints, the classes can further represent individual and joint chance constraints. In general, the stochastic constraints,  $g_i(\mathbf{x}, \xi)$  and  $h_i(\mathbf{x}, \xi)$  can have one of the two mathematical structures

$$\begin{cases} \zeta_1(\mathbf{x}) + \tau\xi & \zeta_1 : \mathbb{R}^n \rightarrow \mathbb{R}, \tau \in \mathbb{R} \\ \zeta_2(\mathbf{x}, \xi) & \zeta_2 : \mathbb{R}^{n+m} \rightarrow \mathbb{R} \end{cases}$$

#### *Variables*

The class for the variables, can represent pure integer  $\{0, 1\}^n$ , general integer  $Z^n$ , continuous  $\mathbb{R}^n$ , and mixed integer  $(\{0, 1\}^{n_1} \cup Z^{n_2} \cup \mathbb{R}^{n_3}, n = n_1 + n_2 + n_3)$  decisions.

The features are supported by the framework are summarised in the table 1.

	Problem components				
	Variables	Constraints		Objective	
Type	continuous	deterministic		deterministic	
	integer	stochastic	expectation	stochastic	expectation
	mixed integer		ICC & JCC		probabilistic
	binary				
	linear				
	non-linear				

Table 1

The features supported by of the framework.

### Computation of the constraints and the objective functions

The random vectors follow a multinomial distribution based on the distribution of the individual random variables. An approximate sample space  $\Omega$  ( $= \{\xi^1, \xi^2, \xi^i \dots \xi^{|\Omega|}\}$ ) is constructed at each simulation. The random vectors  $\xi^i$ 's ( $= \{\xi_1^i, \xi_2^i, \dots \xi_m^i\}$ ) are identically distributed and independent following the underlying probability distribution. The approximate value of the stochastic functions are computed using the sampled random vectors. Let  $\mathbf{x}^*$  be the solution to the problem,  $P_{2SP-Anti}$ . The expected value constraints, equation 5, are computed as

$$E_{\xi}[\cdot] = \begin{cases} \frac{\sum_{j=1}^{|\Omega|} g_i(\mathbf{x}^*, \xi^j)}{|\Omega|}, & \forall g_i \in \bar{G}^s \\ \frac{\sum_{j=1}^{|\Omega|} h_i(\mathbf{x}^*, \xi^j)}{|\Omega|}, & \forall h_i \in \bar{H}^s. \end{cases}$$

The value of the probabilistic constraints is the frequency with which the current solution satisfies the constraints. Let  $n_{\hat{g}_{i,r}^s}$  be the number of random vectors that satisfy the probabilistic inequality constraint given in equation 10.

Then the stochastic inequality constraints are evaluated as

$$Pr[\cdot] = \frac{n_{\hat{g}_i}}{|\Omega|}$$

In the case of the *V-model* we expand the objective as  $E_{\xi}[f^2(\mathbf{x}, \xi)] - E_{\xi}^2[f(\mathbf{x}, \xi)]$  and compute the two terms, that is  $E_{\xi}[\cdot]$  and  $E_{\xi}^2[\cdot]$  separately.

## 5.2 GA Controls

We use a steady-state GA that uses ‘overlapping’ populations in which a proportion of the population,  $p_{repl}$ , is replaced in each generation. Newly generated offspring are added to the population, then the worst individuals are destroyed. The new offspring may or may not make it into the population, depending on whether they are better than the worst in the population. Therefore,  $1 - p_{repl}$  proportion of the individuals of one generation are retained in the consecutive generation and the remaining proportion of the population are replaced using the genetic operators. The individuals in the population are represented as real arrays where the arrays represent the solution vector to the optimisation problem. The cross-over operator respects the structure of the solution vector by crossing components with the same characteristics (that is discrete with discrete and continuous with continuous).

Let

- $V$  denote the set of individuals in the population,
- a given individual in the population is denoted by  $v$ ,
- $K$  be the population size,
- $M$  be the total number of generations that the population evolves, and
- $m$  denote a given generation.

By our definition,  $v$  represents a potential solution to optimisation problem, therefore the dimension of  $v$  is  $n$ . The  $m^{th}$  generation of the population is given by  $V_m$  and the  $k^{th}$  ( $k = 1 \dots K$ ) individual in the population is given by  $v_k$ . Determining the correct population size in GA is important [24,25]. We use population size and number of generations that vary linearly with number of variables,  $K = 10 \times n$  and  $M = 20 \times K$ .

Table 2 shows the values for the control parameters for the Genetic Algorithm. The salient features of the GA are set out below.

### **Initialisation**

The individuals in the population are initialised using uniform random numbers that generate values within the pre-specified interval. On the otherhand, there is a feature wherein, any approximate solution to the given problem can be used to initialise the population.

### **Selection**

We use tournament selection in which two individuals in the population are selected based upon the magnitude of their fitness score with respect to the rest of the population. Then, of these two individual, we select the one with the higher score.

### **Crossover**

We use one-point crossover, with probability  $p_{cross}$ , in order to mate two individuals in the population.

Parameter	Value
Genetic Algorithm	Steady State
Size of the population (K)	$10 \times n$
Number of generation (M)	$20 \times K$
Chromosome representation	floating point
Probability of crossover ( $p_{cross}$ )	0.7
Probability of mutation ( $p_{mut}$ )	0.1
Proportion replaced ( $p_{repl}$ )	0.5

Table 2

Control parameters for the Genetic Algorithm.

### ***Mutation***

We use Gaussian mutator, with probability  $p_{mut}$ , to mutate values of an individual.

***Stopping criteria*** We have terminated the algorithm after  $M$  generations.

### ***Penalty Function***

We construct a parameter-less penalty function for constrained optimisation problems. This penalty function measures the violations of constraints by the individuals in the population. We define a non-negative function  $\rho(v, i)$ , where  $v \in V$  and  $i \in \Gamma$ , and  $\rho(v, i) : V \times (\Gamma) \rightarrow [0, \infty]$ . The penalty function is computed as follows:

$$\rho(v, i) = \begin{cases} \max\{0, \alpha - Pr[\cdot]\} & \text{for constraints of the type } Pr[\cdot] \geq \alpha \\ \max\{0, Pr[\cdot] - \alpha\} & \text{for constraints of the type } Pr[\cdot] \leq \alpha \\ \max\{0, E_\xi[g(\mathbf{x}, \xi)]\} \\ \max\{0, |E_\xi[h(\mathbf{x}, \xi)]|\} \\ \max\{0, g(\mathbf{x})\} \\ \max\{0, |h(\mathbf{x})|\} \end{cases}$$

Let  $\rho_{max}(i) = \max\{\rho(v, i) \mid \forall v \in V_m\}$ , be magnitude of the maximum violation of the  $i^{th}$  constraint for individuals in the current generation.

### ***Scoring Functions***

We define two scoring functions called the *feasibility scoring function* and the *optimality scoring function*. These scoring function provides scores to the individuals in the population based on level of feasibility and optimality respectively.

#### ***Feasibility scoring function***

In order to construct the feasibility scoring function we first define the degree

of constraint satisfaction. The degree of constraint satisfaction,  $\delta(v, i)$ , for an individual measures the relative magnitude of the violation of the constraints with respect to the other individuals in the same generation. The non-negative function,  $\delta(v, i)$ , is constructed using the penalty function as follows:

$$\delta(v, i) = \begin{cases} 1 & \text{if constraint } i \text{ is satisfied} \\ \frac{\rho_{max}(v, i) - \rho(v, i)}{\rho_{max}(v, i)}, & \text{otherwise} \end{cases} \quad (13)$$

therefore  $\delta(v, i) : V \times \Gamma \rightarrow [0, 1]$ . The plot of the degree of feasibility satisfac-

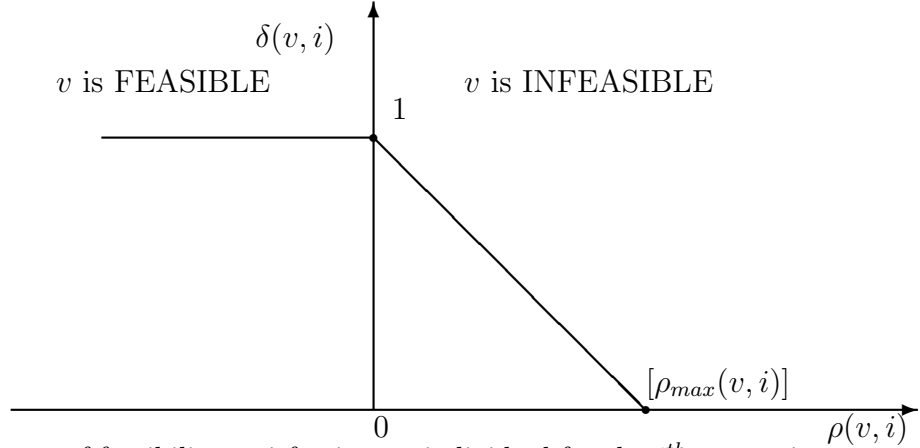


Fig. 2. Degree of feasibility satisfaction an individual for the  $i^{th}$  constraint.

tion for the individuals in the population is illustrated in the figure 2.

We define two alternative variants of the feasibility scoring functions. These are the additive  $\nu(\cdot)_{\oplus} : \Re^n \rightarrow [0, 1]$ , and the multiplicative  $\nu(\cdot)_{\otimes} : \Re^n \rightarrow [0, 1]$  scoring functions. The additive function is defined as

$$\nu_{\oplus}(v) = \frac{\sum_{i \in \Gamma} \delta(v, i)}{\Gamma}, \quad (14)$$

and the multiplicative penalty is defined as

$$\nu_{\otimes}(v) = \prod_{i \in \Gamma} \delta(v, i). \quad (15)$$

*Theorem* The scoring functions (14 & 15)

- (1) give a score of 1 to an individual representing a feasible solution,
- (2) an infeasible individual would get a score in the interval  $[0, 1]$ .

*Proof*

- (1) Consider an individual  $v^*$  such that  $\delta(v^*, i) = 1 \forall i \in \Gamma$ . By our definition in equation 13,  $v^*$  is a feasible solution to the optimisation problem. Therefore

$$v_{\oplus} = \frac{\Gamma}{\Gamma} = 1$$

and

$$v_{\otimes} = 1 \times 1 \dots \times 1 \Gamma \text{ times} = 1$$

- (2) Consider an individual that is infeasible solution,  $\hat{v}$ , moreover it is the most infeasible solution, that is  $\rho(\hat{v}, i) = \rho_{max}(i) \forall i$ . Then  $\delta(\hat{v}, i) = 0 \forall i \in \Gamma$  and  $v_{\oplus} = v_{\otimes} = 0$ .

Now consider an individual  $\tilde{v}$  that is feasible for some constraints and infeasible for others. Since,  $0 \leq \delta(\tilde{v}, i) < 1$  for the infeasible constraints,

$$v_{\oplus} < 1$$

and

$$v_{\otimes} < 1.$$

#### *Optimality scoring function*

Let  $\gamma(v)$  denote the objective function value for a given individual  $v$ . Let  $\gamma_{max}$  denote the maximum and  $\gamma_{min}$  the minimum objective values in the current generation. Define the degree of optimality satisfaction,  $\theta(v)$ , as follows

$$\theta(v) = \begin{cases} \frac{\gamma(v)}{\gamma_{max}} & \text{for maximisation problems} \\ \frac{\gamma_{min}}{\gamma(v)} & \text{for minimisation problems} \end{cases}$$

where  $\theta(v) : V \rightarrow (0, 1]$ .  $\theta(v)$  is a measure of the objective value of an individual with respect to the best in the current population. The least optimal individual would get a score in the interval (0,1), based on the distance of its value from the best objective value. The function  $\theta(v)$  is not as stringent as  $\nu_*(v)$ ; whereas the most infeasible solution would get a value of 0 and therefore would be eliminated in the next generation of the population. The least optimal solution, however, always gets non-zero value. This is in order to retain individuals (for mating) that are feasible but need not necessary be optimal.

#### ***Fitness Function***

The fitness function is defined as  $\Psi(\nu_*(v), \theta(v)) : [0, 1] \times (0, 1] \rightarrow [0, 1]$  and is computed as the product of the feasibility and the optimality scoring functions as follows:

$$\Psi(\nu_*(v), \theta(v)) = \nu_*(v) \times \theta(v) \tag{16}$$

where  $*$  could be either  $\oplus$  or  $\otimes$ . The fitness function defined by equation 16 provides equal importance to both the feasibility and the optimality. This fitness function is extended by varying the relative importance of the two

Parameter	Value
$ \Omega $	300
Number of runs (N)	100

Table 3

Control parameters for the Simulation.

scoring functions over the generations. This extended fitness function is defined as

$$\Psi_{\lambda}(\nu_*(v), \theta(v)) = \nu_*(v)^{(1-\lambda_m)} \times \theta(v)^{\lambda_m} \quad (17)$$

where  $m = 1 \dots M$ . The value of  $\lambda$  is bounded to lie within the interval  $[\lambda_{min}, \lambda_{max}]$ . The value of  $\lambda$  is updated as

$$\lambda_{m+1} = \lambda_m - \frac{\lambda_{\Delta}}{M} \quad (18)$$

where  $\lambda_1 = \lambda_{max}$  and  $\lambda_{\Delta} = \frac{(\lambda_{max} - \lambda_{min})}{M}$ .

During the initial generations we are willing to accept infeasible solutions, as these infeasible solutions would help us form a bridge between the (possible disconnected) feasible spaces, hence we provide higher weight to the degree of optimality satisfaction  $\theta(v)$ . However, we progressively increase the importance for feasible solutions by providing less weight to the degree of optimality and more weight to the degree of feasibility. Our intention is to explore wider regions of the search space at the beginning. By increasing the penalty proportionally to the generation number, it increases the pressure for the population to generate more feasible solutions later on and hopefully converge.

### 5.3 Simulation

The values of the stochastic functions within an optimisation problem are approximated using Monte-carlo simulation. In order to evaluate a stochastic function for the solution  $\mathbf{x}^*$  corresponding to the given individual  $v$ , we generate an independent random vectors,  $\xi$ , from the underlying probability distribution. The stochastic functions could either be in the objective or in the constraints. Let  $J^s$  denote the number of stochastic functions

$$J^s = \begin{cases} H^s \cup G^s & \text{if the objective is deterministic,} \\ H^s \cup G^s \cup \text{Objective function} & \text{otherwise.} \end{cases}$$

We estimate the value of the stochastic function for an individual  $v$  by performing  $|\Omega|$  such evaluations. Therefore, in every generation, we perform  $|\Omega| \times |J^s|$  function evaluations per individual in the population. Since there are  $K$  individuals in the population, and each population has  $20 \times K$  generations, therefore, in total we generate

$$20 \times |\Omega| \times K^2 \times |J^s| \quad (19)$$

independent random vectors. These random vectors are used to evaluate the stochastic optimisation problem for each of the  $N$  runs of the simulation. Substituting the values of the default values of the control parameters for GA (table 2) and simulation (table 3) into the equation 19, we get

$$6 \times 10^5 \times n^2 \times |J^s|, \quad (20)$$

where  $n$  is the cardinality of the solution vector.

The most simple stochastic problem would involve only one stochastic function ( $|J^s| = 1$ ) and decision variable of dimension 1 ( $n = 1$ ). Therefore, in our framework we would process  $6 \times 10^5$  random variables in order to estimate the solution for the simplest stochastic problem. However, it is extremely difficult for mathematical programming based techniques to process such large number of random variables, unless sampling techniques. Moreover, usually such techniques assume convexity of the feasible region. Therefore, even if one can process stochastic optimisation problems with many random variables using Mathematical programming techniques, the presence of the discrete decision variables would make the problem non-convex, thereby limiting their suitability.

## 6 Computational Results

The performance characteristics of the proposed framework has been investigated extensively on several optimization problems from the literature. We carried out the experiments on a laptop with an AMD Athlon processor of speed 1.40 Ghz and 256 MB RAM and Windows XP operating system.

Using different genetic operators and control parameters, a number of independent runs are executed for each test problem. Table 2 shows the default values for the control parameters for GA, whereas the table 3 shows the default values for the parameters for simulation. Since we are interested in developing in a generic framework, we investigated five test problems (see appendix A) using the default controls. We perform multiple runs on the test problems using the same default parameters. However, we found that fine tuning the



default controls can have significant effect on the quality of the solutions to the problem.

### 6.1 Psuedocode

- Step 1:** Initialise  $K$ ,  $M$ ,  $|\Omega|$ ,  $p_{cross}$ ,  $p_{mut}$ ,  $p_{repl}$ ,  $\lambda_{min}$ ,  $\lambda_{max}$ , and  $l = 1$ .  
**Step 2:** Initialise  $m = 1$ .  
**Step 3:** Generate the initial population  $V_m$   
**Step 4:**  $\forall v \in V_m$   
    Update  $\lambda_m$   
    Calculate  $\rho(v, i)$ ,  $\delta(v, i) \forall i \in \Gamma$  to get the degree of feasibility  
    Calculate  $\nu(v)$ ,  $\theta(v)$  to get value for the feasibility and optimality scoring functions  
    Calculate  $\Phi(\nu_*, \theta)$  to get the fitness value for the individual  
**Step 5:** Perform one-point crossover by selecting and mating two individuals.  
**Step 6:** Perform Gaussian mutation.  
**Step 7:** Replace a fraction the  $p_{repl}$  proportion of individuals.  
**Step 8:** If  $m = M$  go to *Step 9*, Else  $m = m + 1$  go to *Step 4*.  
**Step 9:**  $l = l + 1$   
**Step 10** If  $l = N$  STOP, Else go to *Step 2*

### 6.2 Analysis of the framework

There are several questions that we need to analyse such as:

- (1) Do the representation and the framework provide consistent solutions to the CCP problems ?
- (2) Do the additive and multiplicative feasibility scoring functions have different influence significantly differently ?
- (3) Does the strategy of varying the importance to the feasibility and optimality scoring functions over generations have significant influence over the quality of the solution ?

#### 6.2.1 Consistency of the solution

In order to investigate the representation and the consistency of the solutions, we construct the confidence intervals for the objective values and the solutions (see tables 5 and 6). The results from the tables show that the confidence intervals contain the known optimal values for the test problems.

- (1) *Problem 1 : Fractional programming problem*

This is a deterministic problem was included in the test test due to its

fractional objective function. The objective value and the solution obtained from the computational investigation correspond to the known global optimum.

(2) *Problem 2: The News vendor problem*

The confidence interval of the framework contains the known optimum solution.

(3) *Problem 3: Feeder Mixer vendor problem*

Our solution differs from the one reported in the paper. However, on additional investigation, we found that our solution is feasible. Therefore, we have obtained a solution that is better than the one reported earlier.

(4) *Problem 4: Pension Fund problem*

We obtained tight bounds on the objective value and the solution vector for the three different variants of the pension fund problem.

(5) *Problem 5: Kilosa farmer problem*

The solution to this problem is not available, however, we have verified by solution by checking its feasibility on the equivalent non-linear programming formulation.

### 6.2.2 Impact of the feasibility scoring function

The Analysis of the variance (Anova) tables B.1 & B.2 and B.3 & B.4 compare the effects of the additive and the multiplicative feasibility scoring functions on the objective function values. The tables B.1 & B.2 show the effect on the Newsboy problem, whereas the tables B.3 & B.4 show the analysis on the Pension Fund problem (with individual chance constraints). Since we performed 100 runs the critical value of the Fisher distribution at 95% level of significance is  $F(1,198,0.05) = 3.88$ . Our null hypothesis is that there is no difference between the two feasibility scoring functions. The values at the Anova tables are much higher than 3.88. Therefore even at 5% level of risk, we cannot reject the null hypothesis that  $\mu_{\oplus}()$  and  $\mu_{\otimes}()$  are different.

### 6.2.3 Impact of the varying fitness function

The Analysis of the variance tables B.5 & B.6 and B.7 & B.8 compare the effect of the scoring functions with and without the update of the parameter  $\lambda$  on the objective function value. The tables B.5 & B.6 show the effect for the Newsboy problem, whereas the tables B.7 & B.8 show the effect for the Pension Fund problem. From the tables, we cannot accept the null hypothesis that there is no difference between  $\Psi(\cdot)$  and  $\Psi_{\lambda}(\cdot)$ .

Problem	Average time in seconds			
	Additive Penalty		Multiplicative Penalty	
	with $\lambda$ update	without $\lambda$ update	with $\lambda$ update	without $\lambda$ update
$P_{FP}$	.26	.24	.26	.24
$P_{NB}$	3.53	3.53	3.52	3.52
$P_{FM}$	96.38	96.42	96.39	96.46
$P_{PF-EXP}$	.5	.5	.5	.5
$P_{PF-ICC}$	213.9	213.47	260.37	218.8
$P_{PF-JCC}$	184.82	184.23	191	189.74
$P_{KF}$	49.52	49.55	49.52	49.51

Table 4

Average time for a single run.

Problem	Objective function value			
	Additive Penalty		Multiplicative Penalty	
	95%	99%	95%	99%
$P_{FP}$	2.40 to 2.48	2.39 to 2.50	2.41 to 2.48	2.39 to 2.49
$P_{NB}$	5.38 to 5.44	5.37 to 5.45	5.39 to 5.47	5.38 to 5.49
$P_{FM}$	38.62 to 40.23	38.29 to 40.56	38.04 to 40.36	37.57 to 40.83
$P_{PF-EXP}$	121495 to 130552	119636 to 132411	121210 to 130511	119301 to 132419
$P_{PF-ICC}$	94655.5 to 106052	92317.4 to 108390	91426.7 to 101513	89357.3 to 103583
$P_{PF-JCC}$	96151 to 98639.8	95640.3 to 99150.5	96644.7 to 98256.5	96314 to 98587.2
$P_{KF}$	4.6 to 4.8	4.6 to 4.9	4.4 to 4.6	4.3 to 4.7

Table 5

Confidence intervals on the objective function value using the fitness function  $\Psi_\lambda(\cdot)$ .

Table 7 shows the null hypothesis investigated and conclusions drawn through the alternate tables .

## 7 Discussion and Conclusion

In this paper we have developed and investigated a scheme to represent CCP problems, and have used the framework of GA and simulation to process such problems. Through a combined framework of Genetic Algorithm (GA) and

Problem	Solution vectors	
	Additive Penalty	
	95%	99%
$P_{FP}$	{.94,-.03,-.05} to { 1,.05,.1 }	{.94,-.05,-.08} to {1,.06,.13 }
$P_{NB}$	{48.29} to {50.1}	{47.91} to {50.48}
$P_{FM}$	{0,.01,.59,.29} to {0.02,.06,.66,.39}	{0, 0.01,.57,.26} to {.02,.07,.68,.41}
$P_{PF-EXP}$	{29.15,50.67,138.42} to {37.55,57.37,154.31}	{27.43,49.30,135.16} to {39.28,58.74,157.58}
$P_{PF-ICC}$	{64.60,64.27,89.33} to {73.56,75.13,106 }	{62.76,62.03,85.92} to {75.40,77.36,109.42}
$P_{PF-JCC}$	{65.93,79.30,85.22} to {69.21,83.60,87.78}	{65.26,78.42,84.70} to {69.88,84.48,88.30}
$P_{KF}$	{2.08,2.25} to {2.55,2.52}	{1.99,2.1} to {2.65,2.58}
	Multiplicative Penalty	
	95%	99%
$P_{FP}$	{.94,-.02,-.05} to {1,.04,.10 }	{.92,-.04,-.09} to {1.02,.05,.138 }
$P_{NB}$	{48.31 } to {50.48}	{47.87} to {50.92}
$P_{FM}$	{-0.04,-.04,.54,.30} to {.07,.08,.70,.40}	{-.07,-.07,.5,.29 } to {.1,.11,.73,.42 }
$P_{PF-EXP}$	{30.02,50.80,138.15} to {36.86,56.91,154.45}	{28.62,49.54,134.80} to {38.27,58.17,157.80}
$P_{PF-ICC}$	{65.87,66.44,85.65} to {71.95,72.80,102.16}	{64.62,65.14,82.26} to {73.2,74.1,105.54 }
$P_{PF-JCC}$	{65.85,80.06,85.57} to {68.99,83.19,87.37}	{65.21,79.43,85.21} to {69.63,83.83,87.73 }
$P_{KF}$	{1.99,2.06} to {2.50,2.44}	{1.89,1.99} to {2.6,2.52}

Table 6

Confidence intervals on the decision variables using the fitness function  $\Psi_\lambda(\cdot)$ .

Tables	The Null hypothesis	Conclusion
B.1 & B.3	$\nu_\oplus(v)$ and $\nu_\otimes(v)$ are no different	Cannot reject
B.2 & B.4	$\nu_\oplus(v)^{1-\lambda}$ and $\nu_\otimes(v)^{1-\lambda}$ are no different	Cannot reject
B.5 & B.7	$\nu_\oplus(v)$ and $\nu_\oplus(v)^{1-\lambda}$ are no different	Reject
B.6 & B.8	$\nu_\otimes(v)$ and $\nu_\otimes(v)^{1-\lambda}$ are no different	Reject

Table 7

The Null hypothesis being investigated in the alternate tables.

simulation, we exploit the success of GAs in processing non-linear and non-convex optimisation problems, and the natural use of simulation for generating the stochastic processes of the random variables.

We have defined a parameter-less penalty function for constrained optimisation problems and an adaptive fitness function that are independent of the problem. We tested the performance our approach on several test problems. These test cases have been chosen or constructed to reflect the versatility of our framework in representing the wide variety of CCP formulations. We have found that the alternate controls within GA, such as crossover, mutation, evaluation, can be fine-tuned in order to improve the performance.

The historic success of Genetic algorithms in processing NP-Hard optimisation can be used for processing the Stochastic programming optimisation problem. The advantage of GA is that it can be combined with other techniques to obtain problem-specific solution algorithms. Unlike traditional optimisation methods GA is better at handling integer variables than continuous variables. Moreover, since the risk constraint are non-linear, the success of GAs in non-linear optimisation could be extend to Stochastic non-linear programming. Several researchers have made use of the coarse grain parallel structure of GAs.

This is important for processing the increasing larger instances of Stochastic programming problems arising in practical applications.

Monte Carlo simulation provide an ideal methodology for evaluating the stochastic functions in a Stochastic programming problem. The use of Variance reduction techniques, such as importance sampling, can be used in order to improve the accuracy and efficiency of the Monte-carlo based simulation method.

## A Description of the test problems

We consider one deterministic and four stochastic optimisation problem in order to investigate the performance.

### ***Problem 1 : Fractional programming problem***

This is a deterministic optimisation problem.

$$P_{FP} \quad \text{Max} \quad \frac{3x_1 + x_2 - 2x_3 + 0.8}{2x_1 - x_2 + x_3} + \frac{4x_1 - 2x_2 + x_3}{7x_1 + 3x_2 - x_3}$$

subject to

$$\begin{aligned} x_1 + x_2 - x_3 &\leq 1, \\ -x_1 + x_2 - x_3 &\leq -1, \\ 12x_1 + 5x_2 + 12x_3 &\leq 34.8, \\ 12x_1 + 12x_2 + 7x_3 &\leq 29.1, \\ -6x_1 + x_2 + x_3 &\leq -4.1, \\ x_i &\geq 0, i = 1, 2, 3. \end{aligned}$$

The known global optimum is (1, 0, 0) with objective function of 2.471428.

### ***Problem 2: The News vendor problem***

Let \$c\$ denote the production cost <sup>1</sup> and \$ \pi\$ the selling price. Shortage of newspapers will result in reduced revenue of \$ \pi\$ per paper whereas excess would result in a holding cost \$ h\$.

Let  $\xi(\omega)$ (random variable) denote the demand on a particular day; assume  $\xi(\omega) \sim N(50, 20)$ , that is  $\mu_\xi = 50$ (mean) &  $\sigma_\xi = 20$ (standard deviation).

The news vendor may want neither excess wastage or the shortage of newspapers for the realisation of the random demand,  $\xi(\omega)$ . He, Therefore bounds his wastage and shortage cost to be under a pre-specified limit for say, 90% of the random realisations.

$$\begin{aligned} P_{NB} \quad \text{Max} \quad & \pi x - cx \\ \text{Subject to} \quad & \\ & P\{h(x - \xi) \leq 5\} \geq 0.90 \quad (\text{Wastage Cost}) \quad (\text{A.1}) \\ & P\{\pi(\xi - x) \leq 27\} \geq 0.90 \quad (\text{Shortage Cost}) \quad (\text{A.2}) \end{aligned}$$

Probabilistic constraint (A.1) can be written as,

---

<sup>1</sup> Inorder to avoid ambiguity all the cost and revenue are calculated per newspaper.

$$\begin{aligned}
& Pr\{.2(x - \xi) \leq 5\} \geq 0.90 \\
& Pr\{x - 25 \leq \xi\} \geq 0.90 \\
& Pr\left\{\left(\left(\frac{(\xi - \mu_\xi)}{\sigma_\xi}\right) \geq \frac{(x - 25 - \mu_\xi)}{\sigma_\xi}\right)\right\} \geq 0.90 \\
& 1 - Pr\left\{\left(\left(\frac{(\xi - \mu_\xi)}{\sigma_\xi}\right) \leq \frac{(x - 25 - \mu_\xi)}{\sigma_\xi}\right)\right\} \geq 0.90 \\
& Pr\left\{\left(\left(\frac{(\xi - \mu_\xi)}{\sigma_\xi}\right) \leq \frac{(x - 25 - \mu_\xi)}{\sigma_\xi}\right)\right\} \leq 0.10.
\end{aligned} \tag{A.3}$$

Since  $\xi(\omega) \sim N(\mu_\xi, \sigma_\xi)$ ,  $\left(\frac{(x-25-\mu_\xi)}{\sigma_\xi}\right) \sim N(0,1)$ .

$$\begin{aligned}
& \Phi\left(\frac{(x - 25 - \mu_\xi)}{\sigma_\xi}\right) \leq 0.10 \\
& \left(\frac{(x - 25 - \mu_\xi)}{\sigma_\xi}\right) \leq \Phi^{-1}(0.10) \\
& x \leq \Phi^{-1}(0.10)\sigma_\xi + 25 + \mu_\xi
\end{aligned}$$

$\mu_\xi = 50$  and  $\sigma_\xi = 20$ ,

$$\text{Therefore } x \leq 49.36. \tag{A.4}$$

Similarly, (A.2) can be evaluated as follows

$$\begin{aligned}
& Pr\{.9(\xi - x) \leq 27\} \geq 0.90 \\
& Pr\{\xi \leq 30 + x\} \geq 0.90 \\
& Pr\left\{\left(\left(\frac{(\xi - \mu_\xi)}{\sigma_\xi}\right) \leq \frac{(x + 30 - \mu_\xi)}{\sigma_\xi}\right)\right\} \geq 0.90
\end{aligned} \tag{A.5}$$

$$\begin{aligned}
& \Phi\left(\frac{(x + 30 - \mu_\xi)}{\sigma_\xi}\right) \geq 0.90 \\
& \left(\frac{(x + 30 - \mu_\xi)}{\sigma_\xi}\right) \geq \Phi^{-1}(0.90) \\
& x \geq \Phi^{-1}(0.90)\sigma_\xi - 30 + \mu_\xi \\
& \text{Therefore } x \geq 45.63.
\end{aligned} \tag{A.6}$$

From equations (A.4) and (A.6) we conclude that the news-vendor must publish between 46 to 49 papers per day in order to satisfy the constraints (A.1) and (A.2). Since the objective function  $= (\pi - c)x$  is linear and monotonically

increasing in  $x$ , he will buy 49 newspapers in order to maximise the profit. The profit made will be \$ 5.39, and the wastage and shortage cost will be under the specified limit for at least 90% of the random realisation of the demand.

**Problem 3: Feed Mixer Problem**

Van de Panne and Popp [66] presented a CCP formulation of feed mixer problem. Four materials - barley, oats, sesame flakes and groundnut meal, are to be mixed for cattle-feed. The proportion of the different materials in the mix should satisfy the minimal nutritional requirements for protein and fat.

Let

- $x_1, x_2, x_3, x_4$  denote the proportion of barley, oats, sesame flakes and groundnut meal in the mix,
- $\xi = (\xi_1, \xi_2, \xi_3, \xi_4)$  be the random variable representing the amount of protein in barley, oats, sesame flakes and groundnut meal respectively.

That CCP problem can be written as follows:

$$\begin{aligned}
 P_{FM} \quad \text{Max} \quad & 24.55x_1 + 26.75x_2 + 39.00x_3 + 40.50x_4 \\
 & 2.3x_1 + 5.6x_2 + 11.1x_3 + 1.3x_4 \geq 5, \quad (Fat \text{ constraint}) \\
 & Pr [\xi_1x_1 + \xi_2x_2 + \xi_3x_3 + \xi_4x_4 \geq 21] \geq p, \quad (Protein \text{ constraint}) \\
 & x_1 + x_2 + x_3 + x_4 = 1, \\
 & x_1, x_2, x_3, x_4 \geq 0
 \end{aligned}$$

where

- $p$  (=0.95) equals the probability level with which the Protein constraint should be satisfied,
- $\xi_1, \xi_2, \xi_3$ , and  $\xi_4$ , have normal distributions  $\mathcal{N}(12, 0.53)$ ,  $\mathcal{N}(11.9, 0.44)$ ,  $\mathcal{N}(41.8, 4.5)$ , and  $\mathcal{N}(52.1, 0.79)$  respectively.

The Protein constraint can be re-formulated as :

$$\begin{aligned}
 & Pr \left\{ \sum_{i=1}^4 \xi_i x_i \geq 21 \right\} \geq 0.95 \\
 & Pr \left\{ \frac{\sum_{i=1}^4 \xi_i x_i - E[\sum_{i=1}^4 \xi_i x_i]}{\sqrt{Var[\sum_{i=1}^4 \xi_i x_i]}} \geq \frac{21 - E[\sum_{i=1}^4 \xi_i x_i]}{\sqrt{Var[\sum_{i=1}^4 \xi_i x_i]}} \right\} \geq 0.95 \\
 & Pr \left\{ \frac{\sum_{i=1}^4 \xi_i x_i - E[\sum_{i=1}^4 \xi_i x_i]}{\sqrt{Var[\sum_{i=1}^4 \xi_i x_i]}} \leq \frac{21 - E[\sum_{i=1}^4 \xi_i x_i]}{\sqrt{Var[\sum_{i=1}^4 \xi_i x_i]}} \right\} \leq 0.05
 \end{aligned}$$



Since  $\left( \frac{\sum_{i=1}^4 \xi_i x_i - E \sum_{i=1}^4 \xi_i x_i}{\sqrt{\text{Var} \sum_{i=1}^4 \xi_i x_i}} \right) \sim \text{Normal}(0, 1),$

$$\Rightarrow 21 \leq \Phi^{-1}(0.05) \sqrt{\text{Var} \left[ \sum_{i=1}^4 \xi_i x_i \right]} + E \left[ \sum_{i=1}^4 \xi_i x_i \right].$$

The co-relation coefficient between the random variables,  $\xi_i$  and  $\xi_j$  is zero.

$$\begin{aligned} \text{Therefore } 21 &\leq \Phi^{-1}(0.05) \sqrt{\sum_{i=1}^4 \text{Var}[\xi_i x_i] + E[\sum_{i=1}^4 \xi_i x_i]} \\ 21 &\leq -1.645 \left\{ (0.53x_1)^2 + (0.44x_2)^2 + (4.5x_3)^2 + (0.79x_4)^2 \right\}^{\frac{1}{2}} + \\ &\quad 12.0x_1 + 11.9x_2 + 41.8x_3 + 52.1x_4 \end{aligned} \quad (\text{A.7})$$

The non-linear constraint A.7 is the deterministic representation of the (stochastic) Protein constraint. The resulting non-linear optimisation problem is

$$\begin{aligned} \max \quad & 24.55x_1 + 26.75x_2 + 39.00x_3 + 40.50x_4 \\ & 2.3x_1 + 5.6x_2 + 11.1x_3 + 1.3x_4 \geq 5, \quad (\text{Fat constraint}) \\ & -1.645 \left\{ (0.53x_1)^2 + (0.44x_2)^2 + (4.5x_3)^2 + (0.79x_4)^2 \right\}^{\frac{1}{2}} + \\ & \quad 12.0x_1 + 11.9x_2 + 41.8x_3 + 52.1x_4 \geq 21 \quad (\text{Protein constraint}) \\ & x_1 + x_2 + x_3 + x_4 = 1, \\ & x_1, x_2, x_3, x_4 \geq 0. \end{aligned} \quad (\text{A.8})$$

Van de Panne and Popp solved the non-linear optimisation problem given in equation A.8 using Zoutendijk's algorithm. They obtained the solution  $\{0.6359, 0, 0.3127, 0.0515\}$  with objective function value 29.89 guilders.

#### **Problem 4: Pension Fund problem**

Henrion [26] has done detailed computation investigation on a CCP formulation of a prototype Pension fund problem. The pension fund of a company has to meet its liabilities for the next  $T$  years. The liabilities shall be covered by investing an initial capital  $B$  in  $P$  different types of bonds.

Let,

$T = 15, P = 3,$

$\gamma_i$  denote the cost for bond  $i, i \in P,$

$\beta_j$  denote the payments for the year  $j, j \in T,$

and,  $\alpha_{ij}$  denote the yield per bond of type  $i$  in the year  $j.$

The liquidity constraint is for the year  $j$  :

$$\underbrace{B - \sum_{i=1}^{|P|} \gamma_i x_i}_{\text{cash after buying bonds}} + \underbrace{\sum_{k=1}^j \sum_{i=1}^{|P|} \alpha_{ik} x_i}_{\text{cumulative yields of bonds}} - \underbrace{\sum_{k=1}^j \beta_k}_{\text{cumulative payment}} \geq 0 \quad (\text{A.9})$$

where  $\gamma = \{ 980, 970, 1050 \}$ ,  $\beta = \{ 11000, 12000, 14000, 15000, 16000, 18000, 20000, 21000, 22000, 24000, 25000, 30000, 31000, 31000, 31000 \}$ ,

$$\alpha = \begin{pmatrix} 0 & 0 & 0 \\ 60 & 65 & 75 \\ 60 & 65 & 75 \\ 60 & 65 & 75 \\ 60 & 65 & 75 \\ 60 & 65 & 75 \\ 1060 & 65 & 75 \\ & 65 & 75 \\ & 65 & 75 \\ & 65 & 75 \\ & 1060 & 75 \\ & & 75 \\ & & 75 \\ & & 75 \\ & & 1075 \end{pmatrix}$$

Define  $a_{ij} = \sum_{k=1}^j \alpha_{ik} - \gamma_i$ ,  $b_j = \sum_{k=1}^j \beta_k - B$ . The liquidity constraint for the terminal time-period could then be re-written as

$$\sum_{i=1}^{|P|} a_{i,T} x_i - b_T.$$

The deterministic optimisation problem is

$$\begin{aligned}
& \text{Max} \quad \sum_{i=1}^{|P|} a_{i,T} x_i - b_T \\
& \text{Subject to} \\
& \quad \sum_{i=1}^{|P|} a_{ij} x_i \geq b_j \quad j = 1, \dots, T
\end{aligned} \tag{A.10}$$

The optimisation problem  $P_{PF-Exp}$  has an optimal solution  $\{31.1, 55.5, 147.3\}$  with an objective function value of \$ 127,332.

Now, assume that the liability streams are random variables  $\eta_j$ , expected value =  $\beta_j$  (the deterministic payments), and standard deviation =  $500j$  (the standard deviation increases with time). The cumulative payment at the end of  $j$  years,  $\xi_j$ , is calculated as  $\sum_{k=1}^j \eta_k$ . Due to the distribution assumptions on  $\eta_j$ , the random vector  $\xi$  has a multivariate normal distribution.

The optimisation problems with individual chance constraint is

$$\begin{aligned}
P_{PF-ICC} \quad & \text{Max} \quad \sum_{i=1}^{|P|} a_{i,T} x_i \\
& Pr \left\{ \sum_{i=1}^{|P|} a_{ij} x_i \geq \xi_j \right\} \geq p \quad (j = 1, \dots, 15)
\end{aligned} \tag{A.11}$$

For the value of  $p = 0.95$ , problem  $P_{PF-ICC}$  has the solution  $\{62.8, 72.6, 101.1\}$  with objective function value \$ 103,925.

Problem A.11 can be extended by imposing the constraint that the liabilities must satisfied *jointly* for all the time periods with 95% probability.

$$\begin{aligned}
P_{PF-JCC} \quad & \text{Max} \quad \sum_{i=1}^{|P|} a_{i,T} x_i \\
& Pr \left\{ \sum_{i=1}^{|P|} a_{ij} x_i \geq \xi_j \quad (j = 1, \dots, 15) \right\} \geq p.
\end{aligned} \tag{A.12}$$

The problem  $P_{PF-JCC}$  has the solution  $\{65.8, 83.7, 86.2\}$  with objective function \$ 98,160.

### ***Problem 5: Kilosa farmer problem***

A family in Kilosa [55,67] has land in which one can grow maize and sorghum. The yields of the crops are uncertain, partly because of uncertain rainfall conditions. Let

$x_m$  = acreage of maize in hectares;

$x_s$  = acreage of sorghum in hectares;

$\xi_m$  = random yield per hectare of maize (in 100 Kgs);

$\xi_s$  = random yield per hectare of sorghum (in 100 Kgs);  
 $\xi$  = random total rainfall during the growing season (mm);  
 $\epsilon_m$  = white noise in the yield of maize;  
 $\epsilon_s$  = white noise in the yield of sorghum.  
 It is known that

- (1) 100 kgs of maize contains  $2.8 \times 10^5$  Kcal and 6.4 kg of protein
- (2) 100 kgs of sorghum contains  $2.8 \times 10^5$  Kcal and 8 kg of protein

Regression analysis applied to empirical data leads to the following model for the random yields.

$$\begin{aligned}\xi_m &= 0.020\xi - 1.65 + \epsilon_1 \\ \xi_s &= 0.008\xi + 5.92 + \epsilon_2\end{aligned}$$

where  $\xi \sim Normal(515.5, \sqrt{18769})$ ,  $\epsilon_1 \sim Normal(0, \sqrt{100})$ ,  $\epsilon_2 \sim Normal(0, \sqrt{100})$   
 The diet of the family should satisfy the nutritional requirements for calories atleast  $\alpha_c$  % and that of proteins atleast  $\alpha_p$  % of the time. The objective is to minimise the land area used for cultivation  $x_m + x_s$  Therefore the optimisation problem is :

$$P_{KF} \quad \text{Min} \quad x_m + x_s \quad (A.13)$$

$$\begin{aligned}Pr\{2.8\xi_m x_m + 2.8\xi_s x_s \geq 44\} &\geq \alpha_c \\ Pr\{6.4\xi_m x_m + 8.0\xi_s x_s \geq 89\} &\geq \alpha_p.\end{aligned} \quad (A.14)$$

The argument described in the Feeder mixer problem earlier can be applied to convert the above CCP problem to its equivalent non-linear programming formulation. We make an assumption that the correlation between  $\xi_m$  and  $\xi_s$  is zero. The constraint A.13 as

$$\begin{aligned}44 \leq \Phi^{-1}(1 - \alpha_c) \{ (2.8x_m)^2(Var(\xi_m)) + (2.8x_s)^2(Var(\xi_s)) \}^{\frac{1}{2}} + \\ 2.8\mu_m x_m + 2.8\mu_s x_s\end{aligned} \quad (A.15)$$

and constraint A.14 as

$$\begin{aligned}89 \leq \Phi^{-1}(1 - \alpha_p) \{ (6.4x_m)^2(Var(\xi_m)) + (8x_s)^2(Var(\xi_s)) \}^{\frac{1}{2}} + \\ 6.4\mu_m x_m + 8\mu_s x_s\end{aligned} \quad (A.16)$$

where  $\mu_{\xi_m} = 10.31$ ,  $\sigma_{\xi_m} = \sqrt{97.15}$ ,  $\mu_{\xi_s} = 6.61$ ,  $\sigma_{\xi_s} = \sqrt{57.72}$ .

Therefore the equivalent non-linear optimisation problem having 80% reliability for the protein and the calories level is :

$$\begin{aligned}
\text{Min} \quad & x_m + x_s \\
& 44 \leq -.84\{(2.8x_m)^2(107.5) + (2.8x_s)^2(101.2)\} \\
& \quad + 2.8(8.66)x_m + 2.8(4.124)x_s \\
& 89 \leq -.84\{(6.4x_m)^2(107.5) + (8x_s)^2(101.2)\} \\
& \quad + 6.4(8.66)x_m + 8(4.124)x_s
\end{aligned}$$

## B Results of the Analysis of Variance

Source	SS	df	MS	F	P-value
Between strategy	818.91	1	818.91	30.95	0
Within strategy	5238.37	198	26.45		
Total	6057.28	199			

Table B.1

Anova table comparing the effects of the feasibility scoring functions  $\nu_{\oplus}(v)$  and  $\nu_{\otimes}(v)$  on the solution of the Newsboy problem.

Source	SS	df	MS	F	P-value
Between strategy	656.66	1	656.66	29.21	0
Within strategy	4449.78	198	22.47		
Total	5106.45	199			

Table B.2

Anova table comparing the effects of the feasibility scoring functions  $\nu_{\oplus}(v)^{(1-\lambda_l)}$  and  $\nu_{\otimes}(v)^{(1-\lambda_l)}$  on the solution of the Newsboy problem.

Source	SS	df	MS	F	P-value
Between strategy	$10^{12}$	1	$10^{12}$	48155.38	0
Within strategy	$10^9$	198	41287464		
Total	$10^{12}$	199			

Table B.3

Anova table comparing the effects of the feasibility scoring functions  $\nu_{\oplus}(v)$  and  $\nu_{\oplus}(v)$  on the objective value of the Pension fund (ICC) problem.

Source	SS	df	MS	F	P-value
Between strategy	$10^{12}$	1	$10^{12}$	18462.26	0
Within strategy	$10^{10}$	198	$10^8$		
Total	$10^{12}$	199			

Table B.4

Anova table comparing the effects of the feasibility scoring functions  $\nu_{\oplus}(v)^{(1-\lambda_l)}$  and  $\nu_{\otimes}(v)^{(1-\lambda_l)}$  on the objective value of the Pension fund (ICC) problem.

Source	SS	df	MS	F	P-value
Between strategy	12.45	1	12.45	.274	.601
Within strategy	8996.572	198	45.43		
Total	9009.02	199			

Table B.5

Anova table comparing the effects of the fitness functions  $\nu_{\oplus}(v) \times \theta(v)$  and  $\nu_{\otimes}(v) \times \theta(v)$  on the solution of the Newsboy problem.

Source	SS	df	MS	F	P-value
Between strategy	.288	1	.288	.082	.773
Within strategy	691.59	198	3.49		
Total	691.879	199			

Table B.6

Anova table comparing effects of the fitness functions  $\nu_{\oplus}(v)^{(1-\lambda_l)} \times \theta(v)^{\lambda_l}$  and  $\nu_{\otimes}(v)^{(1-\lambda_l)} \times \theta(v)^{\lambda_l}$  on the solution of the Newsboy problem

Source	SS	df	MS	F	P-value
Between strategy	40069152	1	40069152	1.018	.314
Within strategy	10 <sup>9</sup>	198	39330140		
Total	10 <sup>9</sup>	199			

Table B.7

Anova table comparing the effects of the fitness functions  $\nu_{\oplus}(v) \times \theta(v)$  and  $\nu_{\otimes}(v) \times \theta(v)$  on the objective value of the Pension fund (ICC) problem.

Source	SS	df	MS	F	P-value
Between strategy	2549282	1	2549282	.022	.8796
Within strategy	10 <sup>10</sup>	198	10 <sup>8</sup>		
Total	10 <sup>10</sup>	199			

Table B.8

Anova table comparing effects of the fitness functions  $\nu_{\oplus}(v)^{(1-\lambda_l)} \times \theta(v)^{\lambda_l}$  and  $\nu_{\otimes}(v)^{(1-\lambda_l)} \times \theta(v)^{\lambda_l}$  on the objective value of the Pension fund (ICC) problem.

## References

- [1] <http://www.robertz.net>, Robert Davies, Statistics Reserch associates Ltd.
- [2] Aringheri, R.: 2004, *Lecture Notes in Computer Science*, Chapt. Solving Chance-Constrained Program Combining Tabu Search and Simulation, pp. 30–41. Berlin Heidelberg: Springer-Verlag.
- [3] Ariyawansa, K. and A. Felt: 2001, ‘On a new collection of stochastic linear programming test problems’. Preprint at Optimization Online ([www.optimization-online.org/](http://www.optimization-online.org/)).
- [4] Barbosa, H. J. C. and A. C. C. Lemonge: 2003, ‘A new adaptive penalty scheme for genetic algorithms’. *Information Sciences* **156**(3-4), 215–251.
- [5] Beale, E.: 1955, ‘On minimizing a convex function subject to linear inequalities’. *J. Royal Statistical Society Series B*, 173–184.
- [6] Ben-Tal, A. and A. Nemirovski: 2002, ‘Robust optimization methodology and applications’. *Mathematical Programming* **92**(3), 453–480.
- [7] Beraldi, P. and A. Ruszczyński: 2002, ‘A branch and bound method for stochastic integer problems under probabilistic constraints’. *Optimization Methods and Software* pp. 359–382.
- [8] Birge, J.: 1985, ‘Decomposition and partitioning methods for multi-stage stochastic linear programs’. *Operations Research* **33**, 989–1007.

- [9] Birge, J., M. Dempster, H. Gassmann, A. King, and S. Wallace: 1987, 'A standard input format for stochastic linear programs.'. *COAL Newsletter* **17**, 1–20.
- [10] Birge, J. and D. Holmes: 2001, 'A portable Stochastic Programming test set'. <http://users.iems.nwu.edu/~jrbirge/html/dholmes/SPTSlists.html>.
- [11] Charnes, A. and W. Cooper: 1959, 'Chance constrained programming'. *Management Science* **5**, 73–79.
- [12] Costa, L. and P. Oliveira: 2001, 'Evolutionary Algorithms Approach to the Solution of Mixed Integer Non-Linear Programming Problems'. *Computers & Chemical Engineering* **25**, 257–266.
- [13] Dantzig, G.: 1955, 'Linear programming under uncertainty'. *Management Science* **1**, 197–206.
- [14] Deb, K.: 2000, 'An efficient constraint handling method for genetic algorithms'. *Computational methods in applied mechanics and engineering* (186), 311–338.
- [15] Dempster, M. and G. Consigli: 1998, 'The CALM stochastic programming model for dynamic asset-liability management'. In: *World Wide Asset and Liability Modelling*. Cambridge University Press, pp. 464–500.
- [16] Dempster, M., N. H. Pedrón, E. Medova, J. Scott, and A. Semboš: 2000, 'Planning logistic operations in the oil industry.'. *Journal of Operational Research Society* **51**(11), 1271–1288.
- [17] Dentcheva, D., A. Prekopa, and Ruszczyński: 2000, 'Concavity and efficient points of discrete distributions in probabilistic programming'. *Mathematical Programming* **89**, 235–263.
- [18] Dentcheva, D., A. Prekopa, and Ruszczyński: 2001, 'On convex probabilistic programming with discrete distributions'. *Nonlinear Analysis* **47**, 1997–2009.
- [19] Dert, C.: 1995, 'Asset Liability Management for Pension Funds: A Multistage Chance Constrained Programming Approach'. Ph.D. thesis, Erasmus University, Rotterdam, The Netherlands.
- [20] Elmaghraby, S. E., H. Soewandi, and M.-J. Yao: 2001, 'Chance-constrained programming in activity networks: A critical evaluation'. *European Journal of Operational Research* **131**(1), 440–458.
- [21] Eppen, G., R. Martin, and L. Schrage: 1989, 'A Scenario Approach to capacity planning'. *Operational Research* **37**(4), 517–525.
- [22] Ghaoui, L. E. and H. Lebre: 1997, 'Robust solutions to least-square problems with uncertain data matrices'. *SIAM Journal of Matrix Analysis Application* **18**, 10351064.
- [23] Goldberg, D.: 1989, *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, New York.



- [24] Goldberg, D., K. Deb, and J. Clark: 1992, 'Genetic Algorithms, noise and the sizing of population'. *Complex system* **6**, 333–362.
- [25] Harik, G., E. Cantu-Paz, D.E. Goldberg, and B.L. Miller: 1997, 'The gamblers win problem, genetic algorithms and the sizing of populations'. pp. 7–12.
- [26] Henrion, R., 'Chance constrained programming: theory and solution methods'. Weierstrass Institute Berlin, Tenth international conference on Stochastic Programming.
- [27] Hightower, J. and S. Sen: 1996, *Stochastic Decomposition : A Statistical Method for large-scale Stochastic Linear Programming*. University of Arizona: Kluwer Academic Publishers.
- [28] Hillier, F.: 1967, 'Chance-constrained programming with 0-1 or bounded continuous decision variables'. *Management Science* **14**, 34–57.
- [29] Holland, J.: 1992, *Adaptation in Natural and Artificial Systems*. Cambridge: MIT Press.
- [30] Homayouni, Lai, and Qi: 1994, 'Constrained optimisation via Genetic Algorithm'. *Simulation* **62**(4), 242–254.
- [31] Ierapetritou, M., J. Acevedo, and E. Pistikopoulos: 1996, 'An Optimization Approach for Process Engineering Problems under Uncertainty'. *Computers and Chemical Engineering* **20**, 703–709.
- [32] Ishii, H., T. Nishida, and Y. Nanbu: 1978, 'A generalized chance constrained programming problem'. *Journal of Operational Research Society of Japan* **21**(1), 124–145.
- [33] Iwamura, K. and B. Liu: 1996, 'A Genetic algorithm for chance constrained programming'. *Journal of Information and Optimization sciences* **17**(2), 409–422.
- [34] Joines, J. and C. Houck: 1994, 'On the use of non-stationary penalty functions to solve non-linear constrained optimisation problems with Genetic algorithms'. In: *Proceedings of the IEEE*. pp. 579–584.
- [35] Kampas, A. and B. White: 2003, 'Probabilistic programming for nitrate pollution control: Comparing different probabilistic constraint approximations'. *European Journal of Operational Research* **147**(1), 217–228.
- [36] Laporte, G., F. Louveaux, and L. V. Hamme: 1994, 'Exact solution to a location problem with stochastic demands'. *Transportation Science* **28**, 95–103.
- [37] Li, Y.-X. and M. Gen: 1996, 'Nonlinear mixed integer programming problems using genetic algorithm and penalty function'. In: *Proceedings of the 1996 IEEE International Conference on Systems, Man and Cybernetics*, Vol. 4. pp. 2677–2682.
- [38] Michalewicz, Z.: 1994, *Genetic Algorithms + Data Structures = Evolution Programs*. Springer-Verlag, 2 edition.

- [39] Michalewicz, Z. and N. Attia: 1994, 'Evolutionary optimisation of constrained problems'. In: *Proceedings of the third annual conference on evolutionary program, world scientific*. pp. 98–108.
- [40] MirHassani, S., C. Lucas, G. Mitra, E. Messina, and C. Poojari: 2000, 'Computational Solution of Capacity Planning Models under Uncertainty'. *Parallel Computing* **26**, 511–538.
- [41] Mulvey, J. and A. Ruszczyński: 1995, 'A new scenario decomposition method for large-scale stochastic optimisation'. *Operational Research* **43**(3), 477–490.
- [42] Murphy, F., S. Sen, and A. L. Soyster: 1982, 'Electric utility capacity expansion planning with uncertain load forecasts'. *AIIE Transaction* **14**, 52–59.
- [43] Olson, D. and S. Swenseth: 1987, 'A linear approximation for chance constrained programming'. *Journal of Operational Research Society* **38**, 261–267.
- [44] Prékopa, A.: 1971, 'Logarithmic concave measures with application to stochastic programming'. *Acta. Sci. Math(Szeged)* **32**, 301–316.
- [45] Prékopa, A.: 1990, 'Dual method for a one-stage Stochastic Programming problem with random RHS obeying a discrete probability distribution'. *Zeitschrift fur Operations Research* **34**, 441–461.
- [46] Prekopa, A.: 1993, 'Programming under probabilistic constraint and maximizing a probability under constraints'. Rrr 35-93, Center for Operations Research, Rutgers university, P.O. box 5062, New Brunswick, New Jersey.
- [47] Richardson, J., M. Palmer, G. Liepins, and M. Hilliard: 1989, 'Some guidelines for GA with Penalty functions'. In: M. Kaufmann (ed.): *Proceedings of the third ICGA*. pp. 191–197.
- [48] Richardson, J. e. a.: 1989, 'Some Guidelines for Genetic Algorithms with Penalty Functions'. In: *Proc. Of the 3rd Inter. Conf. On Genetic Algorithms*. Los Altos, CA, pp. 191–197.
- [49] Robinson, J. R.: 1988, 'Loaded questions: New approaches to utility forecasting'. *Energy policy* **16**(1), 58–68.
- [50] Robinson, S.: 1996, 'Analysis of sample path optimisation'. *Mathematics of Operations Research* (21), 513–528.
- [51] Rockafellar, R. and R. Wets: 1991, 'Scenario and policy aggregation in optimization under uncertainty'. *Mathematics of Operations Research* **16**(1), 119–147.
- [52] Romisch, W. and R. Schultz: 1991, 'Distribution sensitivity in stochastic programming'. *Mathematical Programming* (50), 197–226.
- [53] Ruszczyński, A.: 2002, 'Probabilistic programs with discrete distributions and precedence constrained knapsack polyhedra'. *Mathematical Programming* (93), 195–215.

- [54] Schoenauer, M. and S.Xanthakis: 1993, 'Constrained GA optimisation'. In: M. Kaufmann (ed.): *Proceeding of the fifth ICGA*. pp. 573–580.
- [55] Schweigman, C.: 1985, 'OR in development countries'. Khartoum University press, Khartoum.
- [56] Sen, S.: 1992, 'Relaxations for probabilistically constrained programs with discrete random variables'. *Operations Research Letters* (11), 81–86.
- [57] Sen, S., R. Doverspike, and S. Cosares: 1994, 'Network planning with random demand.'. *Telecommunication Systems* **3**, 11–30.
- [58] Seppala, Y.: 1971, 'Constructing sets of uniformly tighter linear approximations for a chance-constraint'. *Management Science* **17**, 736–749.
- [59] Seppala, Y. and T. Orpana: 1984, 'Experimental study on the efficiency and accuracy of a chance-constrained programming algorithm'. *European Journal of Operational Research* **16**, 345–357.
- [60] Shapiro, A.: 1991, 'Asymptotic analysis of stochastic programs'. *Annals of Operations Research* (30), 169–186.
- [61] Shapiro, A. and T. H. de Mello: 2000, 'On Rate of Convergence of Monte Carlo Approximations of Stochastic Programs'. *SIAM Journal on Optimization* **11**(1), 70–86.
- [62] Slyke, R. V. and R. J.-B. Wets: 1969, 'L-Shaped linear programs with application to optimal control and stochastic programming'. *SIAM Journal on Applied Mathematics* **17**, 638–663.
- [63] Smith, J.: 1999, 'Optimizing Platform Survivability Using a Chance Constrained Linear Program'. In: *US Army Ground Vehicle Survivability Symposium*. Survivability/lethality analysis directorate, munitions and Platform branch, White sands Missile range, NM.
- [64] Takriti, S., J. Birge, and E. Long: 1996, 'A Stochastic Model for the Unit Commitment Problem'. *IEEE Transactions on Power Systems* **11**, 1497–1508.
- [65] Tomasgard, A., S. Dye, S. Wallace, J. Audestad, and L. Stougie: 1998, 'Modelling aspects of distributed processing in telecommunication networks'. *Annals of Operations Research* **82**, 161–184.
- [66] van de Panne, C.: 1963, 'Minimum-Cost Cattle Feed under Probabilistic Protein Constraints'. *Management Science* **9**(3), 405–430.
- [67] van der Vlerk, M. <http://mally.eco.rug.nl/lnmb/cases.pdf>.
- [68] van der Vlerk, M.: 2003, 'Integrated chance constraints in an ALM model for pension funds'. Technical Report 03A21, University of Groningen, Research Institute SOM (Systems, Organisations and Management).
- [69] Wall, M.: 1996, 'GAlib: A C++ library of Genetic Algorithm components'. version 2.3, Massachusetts of Technology.

- [70] Weintraub, A. and J. Vera: 1991, 'A cutting plane approach for chance constrained linear program'. *Operations Research* **39**(5), 770–776.
- [71] Yahia, M. and A. Daneshmand: 1995, 'A linear approximation method for solving a special class of the chance constrained programming problem'. *European journal of operational research* **80**, 213–225.